



Model-free optimal trajectories in the image space: Application to robot vision control

Youcef Mezouar, François Chaumette

► To cite this version:

Youcef Mezouar, François Chaumette. Model-free optimal trajectories in the image space: Application to robot vision control. IEEE Int. Conference on Computer Vision and Pattern Recognition, CVPR'01, 2001, Kauai, Hawaii, France. pp.1115-1162. inria-00352137

HAL Id: inria-00352137

<https://inria.hal.science/inria-00352137>

Submitted on 12 Jan 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Model-Free Optimal Trajectories in the Image Space: Application to Robot Vision Control

Y. Mezouar and F. Chaumette

IRISA/INRIA

Campus universitaire de Beaulieu 35042 Rennes Cedex, France

e-mail: {ymezouar,chaumette}@irisa.fr

Abstract

Since image-based servoing is a local control solution, it requires the definition of intermediate subgoals in the sensor space when the initial robot position is far away from the desired one. This paper addresses the problem of generating and tracking realistic and optimal smooth trajectories of complex features in the image space. The model of the observed target and the internal camera parameters are assumed to be unknown. First a closed-form smooth collineation path (related to a reference plane) between given starts and end-points is obtained. This path is generated in order to correspond to an optimal camera path. The trajectories of the image features (corresponding to points belonging or not to the reference plane) are then derived and tracked using an image based control.

1 Introduction

Image-based servoing is now a well known local control framework [7], [9]. In this approach, the reference image of the object corresponding to a desired position of the robot is acquired first (during an off-line step) and some image features are extracted. An error is directly measured in the image between the current image and the reference one. The robot motion is then controlled in order to minimize the error (using for example a gradient descent approach). Sometimes, and especially when the displacement to realize is large, image and camera trajectories induced by image-based gradient descent are neither physically valid nor optimal [1]. In this article, we address the problem of finding realistic image trajectories (i.e corresponding to physically valid camera motion) and corresponding to an optimal camera path with respect to minimum acceleration criterion. The obtained trajectories can be used to improve significantly the system behavior since a local control solution works properly for a trajectory following [11].

Only few papers deal with path planning in image space. In [8] a trajectory generator using a stereo system is proposed and applied to obstacle avoidance. An alignment task using intermediate views of the object synthesized by image mor-

phing is presented in [15]. A path planning for a straight-line robot translation observed by a weakly calibrated stereo system is performed in [13]. In previous work [11], we have proposed a potential field-based path planning generator that determines the trajectories in the image of a set of points lying on an unknown target. However, none of them were dealing with optimality issues. In this paper, first a smooth and optimal collineation path is obtained. Then, smooth image features trajectories corresponding to an optimal camera trajectory are computed.

The paper is organized as follow. In Section 2, we recall some properties of the collineation matrix. In Section 3 and Section 4, the cases where the camera is displaced according to a pure rotational and a pure translational motion are studied. In Section 5, the general case is addressed. In Section 6, the optimal collineation path is used to perform the optimal path in the image. The results of path planning are given in Section 7. In Section 8, we show how to use an image-based control approach to track the trajectories in the image space.

2 The Collineation matrix

Consider two views of a scene observed by a camera. A 3-D point \mathcal{X} with homogeneous coordinates $\mathbf{X} = [X \ Y \ Z \ 1]^T$ is projected under perspective projection to a point \mathbf{x} in the first image (with homogeneous coordinates measured in pixel $\mathbf{x} = [x \ y \ 1]^T$) and to a point \mathbf{x}^f in the second image (with homogeneous coordinates measured in pixel $\mathbf{x}^f = [x^f \ y^f \ 1]^T$). It is well known that there exists a projective homography matrix \mathbf{G} related to a virtual plane Π , such that for all points \mathcal{X} belonging to Π , $\mathbf{x} \propto \mathbf{G}\mathbf{x}^f$ ($\Leftrightarrow \alpha \mathbf{x} = \mathbf{G}\mathbf{x}^f$ where α is a scaling factor). The matrix \mathbf{G} , when \mathbf{x} and \mathbf{x}^f are expressed in pixels, is called the collineation matrix. From the knowledge of several matched points, lines or contours [4, 2], it is possible to estimate the collineation matrix. For example, If at least four matched points belonging to Π are known, \mathbf{G} can be estimated by solving a linear system. Else, at least height points (3 points to define Π and 5 outside of Π) are necessary to estimate the collineation matrix by using for example the linearized algorithm proposed in [10]. Assuming that the camera calibra-

tion is known, the Euclidean homography can be computed up to a scalar factor:

$$\mathbf{H} \propto \mathbf{K}^+ \mathbf{G} \mathbf{K} \quad (1)$$

where \mathbf{K} is a non singular matrix containing the intrinsic parameters of the camera and it is given by:

$$\begin{bmatrix} fp_u & -fp_u \cot(\alpha) & u_0 \\ 0 & fp_v / \sin(\alpha) & v_0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{a} & u_0 \\ 0 & v_0 \\ 0 & 1 \end{bmatrix}$$

where u_0 and v_0 are the pixels coordinates of principal point, f is the focal length, p_u and p_v are the magnifications respectively in the u and v directions and α is the angle between these axes. The Euclidean homography can be decomposed into a rotation matrix and a rank 1 matrix [5]:

$$\mathbf{H} = \mathbf{R} + \frac{\mathbf{b}}{d^f} \mathbf{n}^{fT} \quad (2)$$

where \mathbf{R} and \mathbf{b} represent the rotation matrix and the translation vector between the current and the desired camera frames (denoted \mathcal{F} and \mathcal{F}^f respectively), \mathbf{n}^f is the unitary normal to the virtual plane expressed in \mathcal{F}^f and d^f is the distance from Π to the origin of \mathcal{F}^f (see Figure 1). From \mathbf{G} and \mathbf{K} , it is thus possible to determine the camera motion parameters (i.e the rotation \mathbf{R} and the scaled translation $\mathbf{b}_{d^f} = \frac{\mathbf{b}}{d^f}$) and the normal vector \mathbf{n}^f . The ratio ρ between the Z -coordinate and d^f , that we will use in the sequel can also be determined:

$$\begin{cases} \rho = \frac{Z}{d^f} = \frac{r}{\mathbf{n}^T \mathbf{m}} & \text{if } \mathcal{X} \in \Pi \\ \rho = \frac{Z}{d^f} = \beta \frac{\|\mathbf{b}_{d^f}\|}{\|\beta \mathbf{m} - \mathbf{R}\|} & \text{if } \mathcal{X} \notin \Pi \end{cases} \quad (3)$$

where $\mathbf{m} = \mathbf{K}^+ \mathbf{x}$, $\mathbf{n} = \mathbf{R} \mathbf{n}^f$, $r = 1 + \mathbf{n}^{fT} \mathbf{R} \mathbf{b}_{d^f}$, $\beta = \frac{\|\mathbf{b}_{d^f} \times \mathbf{R} \mathbf{m}\|}{\|\mathbf{b}_{d^f} \times \mathbf{m}\|}$. If the camera is not perfectly calibrated (i.e $\hat{\mathbf{K}}$ is used instead of \mathbf{K}), then, the parameters which can be estimated are [10]:

$$\hat{\mathbf{R}} = \delta \mathbf{K} \mathbf{R} \delta \mathbf{K}^+ \quad (4)$$

$$\hat{\mathbf{n}}^{fT} = \frac{\mathbf{n}^{fT} \delta \mathbf{K}^+}{\|\mathbf{n}^{fT} \delta \mathbf{K}^+\|} \quad (5)$$

$$\hat{\mathbf{b}}_{d^f} = \|\mathbf{n}^{fT} \delta \mathbf{K}^+\| \delta \mathbf{K} \mathbf{b}_{d^f} \quad (6)$$

where $\delta \mathbf{K} = \hat{\mathbf{K}}^+ \mathbf{K}$. In the sequel we assume that given an initial image and a desired image of the scene, some image features (points, lines, contours) can be extracted and matched. From the extracted image features, the collineation matrix at time $t = 0$, \mathbf{G}_0 , can be computed. Note also that, when the desired configuration is reached (at time $t = 1$) the collineation matrix is proportional to the identity matrix: $\mathbf{G}^f \propto \mathbf{I}$. In the next sections, we consider the problem of finding a smooth path of the collineation matrix between \mathbf{G}_0 and \mathbf{G}^f corresponding to an optimal camera path with respect to criterion which will be specified in the sequel.

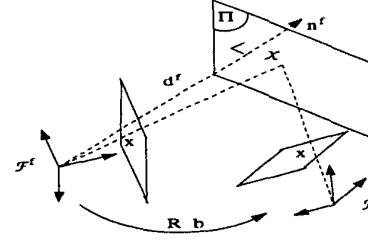


Figure 1: Geometry of two views

3 Pure Rotational motion

In this section, we address the problem of finding an optimal closed form smooth path of the collineation when the collineation is a matrix similar (up to a scalar factor) to a rotation matrix, that is:

$$\mathbf{G} \propto \mathbf{K} \mathbf{R} \mathbf{K}^+ \quad (7)$$

We begin with a brief review of necessary background on the Special Orthogonal group $SO(3)$.

3.1 Brief Review of $SO(3)$

The group $SO(3)$ is the set of all 3×3 real orthogonal matrices with unit determinant and it has the structure of a Lie group. On a Lie group, the space tangent to the identity has the structure of a Lie algebra. The Lie algebra of $SO(3)$ is denoted by $so(3)$. It consists of the 3×3 skew-symmetric matrices, so that the elements of $so(3)$ are matrices of the form:

$$[\mathbf{r}] = \begin{bmatrix} 0 & -r_3 & r_2 \\ r_3 & 0 & -r_1 \\ -r_2 & r_1 & 0 \end{bmatrix}$$

One of the main connections between a Lie group and its Lie algebra is the exponential mapping. For every $\mathbf{R} \in SO(3)$, there exists at least one $[\mathbf{r}] \in so(3)$ such that $e^{[\mathbf{r}]} = \mathbf{R}$ with (Rodriguez formula):

$$\mathbf{R} = e^{[\mathbf{r}]} = \mathbf{I} + \frac{\sin \|\mathbf{r}\|}{\|\mathbf{r}\|} [\mathbf{r}] + \frac{1 - \cos \|\mathbf{r}\|}{\|\mathbf{r}\|^2} [\mathbf{r}]^2 \quad (8)$$

where $\|\mathbf{r}\|$ is the standard Euclidean norm. Conversely, if $\mathbf{R} \in SO(3)$ such that $\text{Trace}(\mathbf{R}) \neq -1$ then:

$$[\mathbf{r}] = \log(\mathbf{R}) = \frac{\phi}{2 \sin \phi} (\mathbf{R} - \mathbf{R}^T) \quad (9)$$

where ϕ satisfies:

$$\phi = \|\mathbf{r}\| = \arccos \left(\frac{1}{2} (\text{Trace}(\mathbf{R}) - 1) \right) \quad (10)$$

If $\text{Trace}(\mathbf{R}) = -1$, $\log(\mathbf{R})$ can be obtained noticing that $\mathbf{r} = \pm \pi \mathbf{u}$ where \mathbf{u} is a unit length eigenvector of \mathbf{R} associated with the eigenvalue 1. Another important connection between $so(3)$ and $SO(3)$ involves angular velocities.

If $\mathbf{R}(t)$ is a curve in $SO(3)$, then $\dot{\mathbf{R}}\mathbf{R}^T$ and $\mathbf{R}^T\dot{\mathbf{R}}$ are skew-symmetric, and hence element of $so(3)$. The element \mathbf{w} of $so(3)$ such that:

$$[\mathbf{w}] = \mathbf{R}^T\dot{\mathbf{R}} \quad (11)$$

corresponds to the angular velocity of the rigid body.

3.2 Optimal collineation trajectories

We consider the problem **PR** of finding a path of the collineation corresponding to the minimum acceleration problem, that is:

$$\text{Find } \mathbf{G}(t) \text{ minimizing } J_1 = \int_0^1 \dot{\mathbf{w}}\dot{\mathbf{w}}^T dt$$

subject to $[\mathbf{w}] = \mathbf{R}^T\dot{\mathbf{R}}$, $\mathbf{G} \propto \mathbf{K}\mathbf{R}\mathbf{K}^+$ and with boundary conditions $\mathbf{G}(0) = \mathbf{G}_0$, $\mathbf{G}(1) = \mathbf{I}$, $\mathbf{w}(0) = \mathbf{0}$ and $\mathbf{w}(1) = \mathbf{0}$. Let us note that the boundary conditions of **PR** are verified if $\mathbf{R}(0) = \mathbf{R}_0$, $\mathbf{R}(1) = \mathbf{I}$, $\mathbf{w}(0) = \mathbf{0}$ and $\mathbf{w}(1) = \mathbf{0}$ (**rotational problem**). Note also that the considered boundary conditions are particularly important since they are indeed the desired boundary conditions in visual servoing applications.

The solution of **PR** and a nice property of the solution are given by the following proposition.

Proposition 1: The optimal path of the collineation matrix is given by:

$$\mathbf{G}(t) \propto \mathbf{G}_0 \Gamma(\mathbf{g}_0, q(t)) \quad (12)$$

where $q(t) = -2t^3 + 3t^2$ and:

$$\Gamma = \mathbf{I} + \frac{\sin(\|\mathbf{g}_0\|q(t))}{\|\mathbf{g}_0\|} \{\mathbf{g}_0\} + \frac{1 - \cos(\|\mathbf{g}_0\|q(t))}{\|\mathbf{g}_0\|^2} \{\mathbf{g}_0\}^2 \quad (13)$$

with:

$$\phi = \|\mathbf{g}_0\| = \text{Arcos} \left(\frac{1}{2} D_0^{-1/3} T_0 - 1 \right) \quad (14)$$

$$\{\mathbf{g}_0\} = \frac{\|\mathbf{g}_0\|}{2 \sin \|\mathbf{g}_0\|} \left(D_0^{1/3} \mathbf{G}_0^+ - D_0^{-1/3} \mathbf{G}_0 \right) \quad (15)$$

where $D_0 = \det(\mathbf{G}_0)$, $T_0 = \text{Trace}(\mathbf{G}_0)$.

The collineation path given by (12) is independent of the internal camera parameters.

Proof: The solution of the rotational problem is given by [12]:

$$\mathbf{R}(t) = \mathbf{R}_0 \mathbf{e}^{[\mathbf{r}_0]q(t)} \quad (16)$$

where $[\mathbf{r}_0] = \log(\mathbf{R}_0^T)$ and $q(t) = -2t^3 + 3t^2$. In the case of a pure rotational motion, it is easy to show from (7) that:

$$\begin{aligned} \mathbf{K}\mathbf{R}\mathbf{K}^+ &= D^{-1/3} \mathbf{G} \\ \text{Trace}(\mathbf{R}) &= D^{-1/3} T \end{aligned} \quad (17)$$

where $D = \det(\mathbf{G})$ and $T = \text{Trace}(\mathbf{G})$. According to (7) and (16) we have:

$$\mathbf{G}(t) \propto \mathbf{K}\mathbf{R}_0 \mathbf{e}^{[\mathbf{r}_0]q(t)} \mathbf{K}^+ = \mathbf{K}\mathbf{R}_0 \mathbf{K}^+ \mathbf{K} \mathbf{e}^{[\mathbf{r}_0]q(t)} \mathbf{K}^+$$

This equation can be rewritten as $\mathbf{G}(t) \propto \mathbf{G}_0 \Gamma$, where $\Gamma = \mathbf{K} \mathbf{e}^{[\mathbf{r}_0]q(t)} \mathbf{K}^+$ and according to (8), we obtain:

$$\Gamma = \mathbf{I} + \frac{\sin(\|\mathbf{r}_0\|q(t))}{\|\mathbf{r}_0\|} \mathbf{K}[\mathbf{r}_0] \mathbf{K}^+ + \frac{1 - \cos(\|\mathbf{r}_0\|q(t))}{\|\mathbf{r}_0\|^2} \mathbf{K}[\mathbf{r}_0]^2 \mathbf{K}^+$$

With $\|\mathbf{g}_0\| = \|\mathbf{r}_0\|$ and $\{\mathbf{g}_0\} = \mathbf{K}[\mathbf{r}_0] \mathbf{K}^+$, the previous equation can be rewritten as (13), and (see (9)):

$$\{\mathbf{g}_0\} = \mathbf{K} \log(\mathbf{R}_0^T) \mathbf{K}^+ = \frac{\|\mathbf{g}_0\|}{2 \sin \|\mathbf{g}_0\|} (\mathbf{K}\mathbf{R}_0^T \mathbf{K}^+ - \mathbf{K}\mathbf{R}_0 \mathbf{K}^+)$$

By introducing (17) in the previous equation, we obtain (15). Finally, using (10) and (17), we deduce (14). The initial value of the collineation \mathbf{G}_0 is not affected by errors on intrinsic parameters since it is extracted directly from image data. According to (12), (13), (14) and (15) and noticing that $\{\mathbf{g}_0\}$ and $\|\mathbf{g}_0\|$ depend only on \mathbf{G}_0 the collineation path given by (12) is independent of the camera parameters.

Remark: The path given by the proposition 1 corresponds to a shortest distance path of the rotation matrix (minimal geodesic) with respect to an adequately chosen Riemannian metric on $SO(3)$.

In the next subsection, the case of a pure translational camera motion is studied.

4 Pure translational motion

If the camera motion is a pure translation, the collineation matrix has the following particular form:

$$\mathbf{G} \propto \mathbf{I} + \mathbf{K} \mathbf{b}_d \mathbf{n}^f \mathbf{T} \mathbf{K}^+ \quad (18)$$

In this section, we address the problem **PT** of finding the path of the collineation when it is given by (18) and corresponding to the minimum acceleration problem, that is:

$$\text{Find } \mathbf{G}(t) \text{ minimizing } J_2 = \int_0^1 \dot{\mathbf{v}}^T \dot{\mathbf{v}} dt$$

where \mathbf{v} denotes the time derivative of \mathbf{b} , subject to (18), with boundary conditions $\mathbf{G}(0) \propto \mathbf{G}_0$, $\mathbf{G}(1) \propto \mathbf{I}$, $\mathbf{v}(0) = \mathbf{v}(1) = \mathbf{0}$. Note once again that the boundary conditions of **PT** are verified if $\mathbf{b}(0) = \mathbf{b}_0$, $\mathbf{b}(1) = \mathbf{0}$ and $\mathbf{v}(0) = \mathbf{v}(1) = \mathbf{0}$ (**translational problem**). The considered boundary conditions are indeed the desired boundary conditions in the context of visual servoing.

Proposition 2: The optimal path of the collineation matrix in the sense of **PT** is given by:

$$\mathbf{G}(t) \propto q(t) \mathbf{I} + \frac{(1 - q(t))}{\alpha_0} \mathbf{G}_0 \quad (19)$$

where α_0 is a real solution of the equation:

$$2\alpha^3 - T_0\alpha^2 + D_0 = 0 \quad (20)$$

The optimal smooth trajectories given by (19) is not affected by error on intrinsic parameters.

Proof: Using the classical formalism of optimal control, it is easy to proof that the solution of the translational problem is given by:

$$\mathbf{b}(t) = (1 - q(t))\mathbf{b}_0 \quad (21)$$

By introducing (21) in (18), we obtain:

$$\begin{aligned} \mathbf{G}(t) &\propto \mathbf{I} + \mathbf{K}(1 - q(t))\mathbf{b}_{d'0}\mathbf{n}^{fT}\mathbf{K}^+ \\ &\propto q(t)\mathbf{I} + (1 - q(t))(\mathbf{I} + \mathbf{K}\mathbf{b}_{d'0}\mathbf{n}^{fT}\mathbf{K}^+) \end{aligned}$$

and noticing that $\mathbf{G}_0 = \alpha_0(\mathbf{I} + \mathbf{K}\mathbf{b}_{d'0}\mathbf{n}^{fT}\mathbf{K}^+)$, we deduce (19). From (18), we easily obtain:

$$\begin{cases} \text{Trace}(\mathbf{G}_0) = T_0 = \alpha_0(3 + \mathbf{b}_{d'0}^T\mathbf{n}) \\ \text{Det}(\mathbf{G}_0) = D_0 = \alpha_0^3(1 + \mathbf{b}_{d'0}^T\mathbf{n}) \end{cases}$$

and by combining the previous equations, we deduce that α_0 is a solution of (20).

The path of the collineation matrix given by (19) is independent of the \mathbf{K} -matrix since the initial value of the collineation matrix \mathbf{G}_0 and thus α_0 (refer to (20)) are independent of the camera intrinsic parameters.

Remark: The path given by (19) corresponds to a shortest distance path in \mathbb{R}^3 (straight line) between the initial and desired camera positions.

5 General camera motions

We assume now that the current position of the camera with respect to its desired position is given by the rotation matrix $\mathbf{R}(t)$ and the translation vector $\mathbf{b}(t)$. In this case the collineation matrix is given by $\mathbf{G}(t) \propto \mathbf{K}^+(\mathbf{R} + \mathbf{b}_{d'}\mathbf{n}^{fT})\mathbf{K}$. We denote \mathbf{U} the 6×1 vector $[\mathbf{v}^T \mathbf{w}^T]^T$. We address now the following problem (PC):

$$\text{Find } \mathbf{G}(t) \text{ minimizing } J_3 = \int_0^1 \dot{\mathbf{U}}^T \dot{\mathbf{U}} dt$$

subject to (11), $\mathbf{v} = \dot{\mathbf{b}}$ and with boundary conditions: $\mathbf{G}(0) \propto \mathbf{G}_0$, $\mathbf{G}(1) \propto \mathbf{I}$, $\mathbf{U}(0) = \mathbf{U}(1) = \mathbf{0}_{6 \times 1}$. The boundary conditions are verified if $\mathbf{R}(0) = \mathbf{R}_0$, $\mathbf{b}(0) = \mathbf{b}_0$, $\mathbf{R}(1) = \mathbf{I}$ and $\mathbf{b}(1) = \mathbf{0}$ (**rigid motion problem**). Once again, the considered boundary conditions are particularly important since they are the desired boundary conditions in the context of visual servoing.

The solution of PC and a result about its dependence on

camera calibration are given by the following proposition.

Proposition 3: The optimal path of the collineation matrix in the sense of PC is given by:

$$\mathbf{G}(t) \propto (1 - q(t))\Phi_0 + (\mathbf{G}_0 + \Phi_0)\Gamma(\mathbf{r}_0, t) \quad (22)$$

where:

$$\begin{cases} \Gamma(\mathbf{r}_0, q(t)) = \mathbf{K}\mathbf{e}^{[\mathbf{r}_0]q(t)}\mathbf{K}^+ \\ \Phi_0 = \mathbf{K}\mathbf{b}_{0d'}\mathbf{n}^{fT}\mathbf{K}^+ \end{cases} \quad (23)$$

The path given by (23) is not affected by error on intrinsic error.

Proof: The solution of the rigid motion problem is given by (refer to equations (12) and (21)):

$$\begin{cases} \mathbf{b}(t) = (1 - q(t))\mathbf{b}_0 \\ \mathbf{R}(t) = \mathbf{R}_0\mathbf{e}^{[\mathbf{r}_0]q(t)} \end{cases}$$

According to (1) and (2), the corresponding collineation path is given by:

$$\mathbf{G}(t) \propto \mathbf{K}(\mathbf{R}_0\mathbf{e}^{[\mathbf{r}_0]q(t)} + (1 - q(t))\mathbf{b}_{0d'}\mathbf{n}^{fT})\mathbf{K}^+$$

This path is equivalent to the path given by :

$$\mathbf{G}(t) \propto \mathbf{K}(\mathbf{R}_0\mathbf{e}^{[\mathbf{r}_0]q(t)} + (1 - q(t))\mathbf{b}_{0d'}\mathbf{n}^{fT} + \mathbf{b}_{0d'}\mathbf{n}^{fT}\mathbf{e}^{[\mathbf{r}_0]q(t)} - \mathbf{b}_{0d'}\mathbf{n}^{fT}\mathbf{e}^{[\mathbf{r}_0]q(t)})\mathbf{K}^+$$

and can be rewritten as (22). We note that the matrix of camera internal parameters appears explicitly in the path given by (22). However, the initial collineation matrix \mathbf{G}_0 is not affected by error on intrinsic parameters since it is directly computed from image data. Assume now that the non-singular matrix $\hat{\mathbf{K}}$ is used instead of \mathbf{K} , we have:

$$\begin{cases} \hat{\Phi}_0 = \hat{\mathbf{K}}\hat{\mathbf{b}}_{0d'}\hat{\mathbf{n}}^{fT}\hat{\mathbf{K}}^+ \\ \hat{\Gamma}(\mathbf{r}_0, q(t)) = \hat{\mathbf{K}}\mathbf{e}^{[\hat{\mathbf{r}}_0]q(t)}\hat{\mathbf{K}}^+ \end{cases}$$

By introducing (5) and (6) in $\hat{\Phi}_0$, we obtain:

$$\begin{aligned} \hat{\Phi}_0 &= \hat{\mathbf{K}}\|\mathbf{n}^{fT}\delta\mathbf{K}^+\|\delta\mathbf{K}\mathbf{b}_{d'}\frac{\mathbf{n}^{fT}\delta\mathbf{K}^+}{\|\mathbf{n}^{fT}\delta\mathbf{K}^+\|}\mathbf{K}^+ \\ &= \mathbf{K}\mathbf{b}_{0d'}\mathbf{n}^{fT}\mathbf{K}^+ = \Phi_0 \end{aligned}$$

Furthermore (refer to (9)):

$$[\hat{\mathbf{r}}_0] = \frac{\hat{\phi}}{2 \sin(\hat{\phi})}(\hat{\mathbf{R}}_0^T - \hat{\mathbf{R}}_0) = \hat{\beta}(\hat{\mathbf{R}}_0^T - \hat{\mathbf{R}}_0) \quad (24)$$

where $\hat{\beta} = \frac{\hat{\phi}}{2 \sin \hat{\phi}}$ and $\hat{\phi}$ satisfies $1 + 2 \cos \hat{\phi} = \text{Trace}(\hat{\mathbf{R}}_0)$.

Since $\hat{\mathbf{R}}_0$ is similar to \mathbf{R}_0 (see (4)), $\text{Trace}(\hat{\mathbf{R}}_0) = \text{Trace}(\mathbf{R}_0)$

and thus $\hat{\phi} = \phi$ and $\hat{\beta} = \beta$. By injecting (24) in (8), we obtain :

$$\hat{\Gamma}(\mathbf{r}_0, q(t)) = \hat{\mathbf{K}}(\mathbf{I} + \beta \frac{\sin(\|\mathbf{r}_0\|q(t))}{\|\mathbf{r}_0\|})(\hat{\mathbf{R}}_0^T - \hat{\mathbf{R}}_0) + \beta^2 \frac{1 - \cos(\|\mathbf{r}_0\|q(t))}{\|\mathbf{r}_0\|^2}(\hat{\mathbf{R}}_0^T - \hat{\mathbf{R}}_0)^2 \hat{\mathbf{K}} +$$

Noticing that $\hat{\mathbf{K}}\hat{\mathbf{R}}_0\hat{\mathbf{K}}^+ = \mathbf{R}_0$, we finally obtain $\hat{\Gamma}(\hat{\mathbf{r}}_0, q(t)) = \Gamma(\mathbf{r}_0, q(t))$. Thus, the collineation trajectory given by (22) is independent of the choice of the non singular matrix $\hat{\mathbf{K}}$.

6 Features trajectories in the image

In order to control efficiently a robot using visual data, we have to determine the trajectories of some image features in the image space. More precisely, we want to perform smooth trajectories $\mathbf{s}^*(t) = [x_1^*(t) \ y_1^*(t) \ \dots \ x_n^*(t) \ y_n^*(t)]^T$ of n projected points in the image between a given start point $\mathbf{s}^*(0) = [x_1^*(0) \ y_1^*(0) \ \dots \ x_n^*(0) \ y_n^*(0)]^T$ and a given desired point $\mathbf{s}^*(1) = [x_1^*(1) \ y_1^*(1) \ \dots \ x_n^*(1) \ y_n^*(1)]^T$. We denote $\mathbf{x}_i^*(t) = [x_i^*(t) \ y_i^*(t) \ 1]^T$ the vector of homogeneous coordinates expressed in pixel of the projection of a 3-D point \mathcal{X}_i in the current desired image (at time t). We define vector $\mathbf{h}_i = \alpha_i(t)\mathbf{x}_i^*(t)$. It is well known that for all points (belonging or not to the reference plane):

$$\mathbf{h}_i(t) = \alpha_i(t)\mathbf{x}_i^*(t) = \mathbf{G}(t)\mathbf{x}_i^*(1) + \tau_i\mathbf{e}(t) \quad (25)$$

where $\alpha_i(t)$ is a positive scaling factor depending on time, τ_i is a constant scaling factor null if the target point belongs to Π and $\mathbf{e}(t) = \mathbf{K}\mathbf{b}(t)$ represents the epipole. After that the initial collineation has been estimated, the optimal path of the collineation matrix can be computed as described previously. The initial value of the epipole, $\mathbf{e}(0) = \mathbf{e}_0$, can also be computed directly from image data (i.e, \mathbf{e}_0 is independent of the \mathbf{K} -matrix). Furthermore, it is easy to show (from (21)) that the optimal trajectories of the epipole, with respect to the previously cited criteria, are of the form:

$$\mathbf{e}(t) = (1 - q(t))\mathbf{e}_0 \quad (26)$$

Such trajectories of the epipole are not affected by error on intrinsic parameters. Note also that the scaling factor τ_i is not time dependent and can be computed directly from the initial and desired image data ¹:

$$\tau_i = \text{sign} \left(- \frac{[\mathbf{G}_0\mathbf{x}_i^*(1) \wedge \mathbf{x}_i^*(0)]_1}{[\mathbf{e}_0 \wedge \mathbf{x}_i^*(0)]_1} \right) \frac{\|\mathbf{G}_0\mathbf{x}_i^*(1) \wedge \mathbf{x}_i^*(0)\|}{\|\mathbf{e}_0 \wedge \mathbf{x}_i^*(0)\|}$$

The vector \mathbf{h}_i is not affected by error on intrinsic parameters since $\mathbf{G}(t)$, $\mathbf{e}(t)$ and τ_i ($\forall i \in \{1 \dots n\}$) can be computed without error even if the \mathbf{K} -matrix is unknown. The trajectories of the considered point in the image corresponding to

¹ \mathbf{v}_j denotes the j^{th} components of \mathbf{v}

an optimal camera path can thus also be computed without error, using:

$$x_i^*(t) = \frac{[\mathbf{h}_i(t)]_1}{[\mathbf{h}_i(t)]_3} \quad y_i^*(t) = \frac{[\mathbf{h}_i(t)]_2}{[\mathbf{h}_i(t)]_3} \quad (27)$$

7 Path planning results

Experiments were performed with images acquired by a CCD camera mounted on the effector of a six degree of freedom manipulator. In the first subsection, we construct intermediate views between the initial and the desired images of a target. In the second subsection, we check the robustness of our path planning scheme.

7.1 Synthesis of intermediate view

The intrinsic parameters (\mathbf{K} -matrix) given by the camera manufacturer are used. The camera displacement between initial and desired images (boxed in Figure 2) is very large. The initial collineation matrix is:

$$\mathbf{G}_0 = \begin{bmatrix} 0.4603 & 0.4145 & -597.3284 \\ -0.2476 & 0.6107 & -376.0835 \\ -0.0001 & 0.0006 & -1.5281 \end{bmatrix}$$

Intermediate images synthesized using equation (27) and the solution of the problem PC are given in Figure 2. The corresponding camera trajectory is plotted in Figure 3.

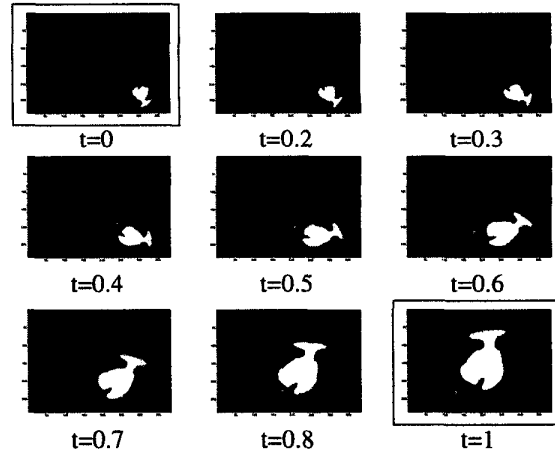


Figure 2: Image trajectories

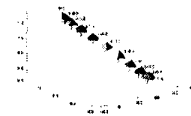


Figure 3: Camera trajectory

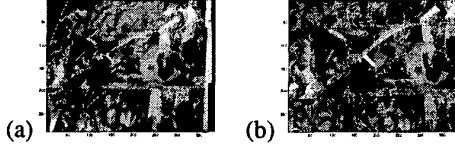


Figure 4: (a) Initial and (b) desired images

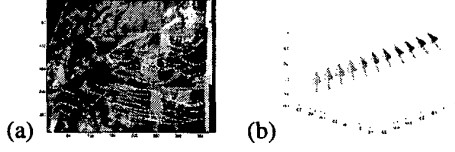


Figure 5: Correct calibration: (a) trajectories in the image and (b) camera trajectory

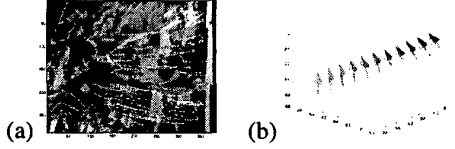


Figure 6: Bad calibration: (a) trajectories in the image and (b) camera trajectory

7.2 Effect of calibration errors

In practice, we use the algorithm proposed in [5] to estimate the motion parameters given by relations (4), (5) and (6). This algorithm was designed for the case where the camera calibration is known. In order to check the robustness with respect to camera calibration errors, we carried out three experiments. In these experiments, we plan the trajectories between an initial and desired images of interesting points (obtained using the Harris detector [6]). These points are matched using the algorithm proposed in [16]. They are represented by white marks in the images corresponding to the initial and desired camera positions given in Figures 4(a) and 4(b) respectively. As far as calibration is concerned, three different sets of parameters have been used :

1. correct calibration : the correct intrinsic parameters have been used (see Figure 5);
2. bad calibration : an error of 50 % has been added on the intrinsic parameters (see Figure 6) ;
3. very bad calibration : an error of 90 % has been added on the intrinsic parameters (see Figure 7).

As can be seen in the Figures 5(a), 6(a) and 7(a) the planned trajectories in the image are closed in all cases. The corresponding camera trajectories given in the Figures 5(b), 6(b) and 7(b) are also similar. That confirms the robustness of the path planning scheme with respect to calibration errors.

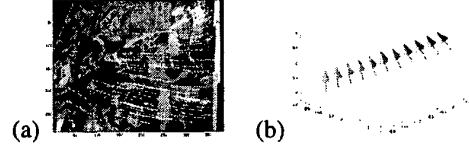


Figure 7: Very bad calibration: (a) trajectories in the image and (b) camera trajectory

8 Application to visual servoing

8.1 Control law

To track the trajectories using an image-based control scheme, we use the task function approach introduced by Samson *et al* in [14]. A vision-based task function \mathbf{e} to be regulated to 0 is defined by [3]:

$$\mathbf{e} = \hat{\mathbf{L}}^+(\mathbf{s}(t) - \mathbf{s}^*(t)) \quad (28)$$

The time varying vector $\mathbf{s}^*(t)$ is the desired trajectory of \mathbf{s} computed as previously explained. The matrix \mathbf{L} denotes the interaction matrix related to \mathbf{s} (also called image Jacobian). It links the variation of the visual features with respect to the camera velocity \mathbf{T}_c : $\dot{\mathbf{s}} = \mathbf{L} \mathbf{T}_c$. For a point \mathcal{X} with coordinates $[X Y Z]^T$ in the current camera frame and image coordinates $\mathbf{x} = [x y 1]^T$ in pixels, the interaction matrix $\mathbf{L}(\mathbf{x}, Z)$ related to \mathbf{x} is given by:

$$\mathbf{L}(\mathbf{x}, Z) = \begin{bmatrix} -\frac{1}{Z} & 0 & \frac{m_x}{Z} & m_x m_y & -(1 + m_x^2) & m_y \\ 0 & -\frac{1}{Z} & \frac{m_y}{Z} & (1 + m_y^2) & -m_x m_y & -m_x \end{bmatrix}$$

where $\mathbf{m} = [m_x m_y 1]^T = \mathbf{K}^+ \mathbf{x}$. When \mathbf{s} is composed of the image coordinates \mathbf{x}_i of n points, the corresponding interaction matrix is:

$$\mathbf{L}(\mathbf{s}, \mathbf{Z}) = [\mathbf{L}^T(\mathbf{x}_1, Z_1) \cdots \mathbf{L}^T(\mathbf{x}_n, Z_n)]^T \quad (29)$$

The matrix $\hat{\mathbf{L}}^+$ is the pseudo-inverse of a chosen model of \mathbf{L} . The value of \mathbf{L} at the current desired position is used for $\hat{\mathbf{L}}$, that is $\hat{\mathbf{L}} = \mathbf{L}(\mathbf{s}^*(t), \hat{\mathbf{Z}}^*(t))$, $\hat{\mathbf{Z}}^*(t)$ being the vector containing the Z -coordinates of each considered point in the camera frame. According to (3), this vector can be rewritten as $\hat{\mathbf{Z}}^*(t) = \hat{d}^f \Gamma^*(t)$ where \hat{d}^f is an estimated value of d^f (that is the only parameter has to be introduced "by hand") and $\Gamma^*(t) = [\rho_1^*(t) \cdots \rho_n^*(t)]$. The ratio $\rho_i^*(t)$ can be estimated (when $\hat{\mathbf{K}}$ is used instead of \mathbf{K}) using relations (3):

$$\begin{cases} \hat{\rho}_i^*(t) = \frac{\hat{r}^*(t)}{\hat{\mathbf{n}}^{*T}(t) \hat{\mathbf{m}}_i(t)} & \text{if } \mathcal{X}_i \in \Pi \\ \hat{\rho}_i^*(t) = \hat{\beta} \frac{\|\hat{\mathbf{b}}_{d^*}\|}{\|\hat{\beta} \hat{\mathbf{m}} - \hat{\mathbf{R}}\|} & \text{if } \mathcal{X}_i \notin \Pi \end{cases} \quad (30)$$

In order to introduce the parameters computed during the path planning step, and after some manipulation, relations

(30) can be rewritten as follow:

$$\begin{cases} \hat{\rho}_i^*(t) = \frac{\det(\mathbf{G}^*(t)) \det(\mathbf{G}_0 - \Phi_0)^{-2/3}}{(\hat{\mathbf{K}}^+ (\mathbf{G}_0 - \Phi_0) \Gamma \hat{\mathbf{K}} \hat{\mathbf{n}}^f)^T \hat{\mathbf{K}}^+ \mathbf{x}_i^*(t)} & \text{if } \mathcal{X}_i \in \Pi \\ \hat{\rho}_i^*(t) = \frac{(1-q(t))\hat{\beta}(t)\|\hat{\mathbf{K}}^+ \Phi_0 \hat{\mathbf{K}} \hat{\mathbf{n}}^f\|}{\|\hat{\beta}(t)\hat{\mathbf{K}}^+ \mathbf{p}_i^* - \hat{\mathbf{K}}^+ (\mathbf{G}_0 - \Phi_0) \Gamma^*(t) \mathbf{p}_i^*(t)\|} & \text{if } \mathcal{X}_i \notin \Pi \end{cases}$$

where:

$$\hat{\beta}(t) = \frac{\|[\hat{\mathbf{K}}^+ \Phi_0 \hat{\mathbf{K}} \hat{\mathbf{n}}^f]_{\times} (\mathbf{G}_0 - \Phi_0) \Gamma^*(t) \mathbf{x}_i^*(1)\|}{\|[\hat{\mathbf{K}}^+ \Phi_0 \hat{\mathbf{K}} \hat{\mathbf{n}}^f]_{\times} \hat{\mathbf{K}}^+ \mathbf{x}_i^*(t)\|}$$

In practice, we use these last relations to estimate $\rho_i^*(t)$ since the analytical form of all these components are known.

An exponential decay of \mathbf{e} toward 0 can be obtained by imposing $\dot{\mathbf{e}} = -\lambda \mathbf{e}$ (λ being a proportional gain), the corresponding control law is:

$$\mathbf{T}_c = -\lambda \mathbf{e} - \frac{\partial \mathbf{e}}{\partial t}$$

where \mathbf{T}_c is the camera velocity sent to the robot controller. If the target is known to be motionless, we have $\frac{\partial \mathbf{e}}{\partial t} = -\hat{\mathbf{L}}^+ \frac{\partial \mathbf{s}^*}{\partial t}$ and the camera velocity can be rewritten:

$$\mathbf{T}_c = -\lambda \mathbf{e} + \hat{\mathbf{L}}^+ \frac{\partial \mathbf{s}^*}{\partial t}$$

where the term $\hat{\mathbf{L}}^+ \frac{\partial \mathbf{s}^*}{\partial t} = \hat{\mathbf{L}}^+ [\dot{x}_1^*(t) \dot{y}_1^*(t) \dots \dot{x}_n^*(t) \dot{y}_n^*(t)]^T$ allows to compensate the tracking error. More precisely, we have from (25):

$$\dot{\mathbf{x}}_i^*(t) = \frac{1}{\alpha_i(t)} [\dot{\mathbf{G}}(t) \mathbf{x}_i^*(1) + \beta_i \dot{\mathbf{e}}(t) - \dot{\alpha}_i(t) \mathbf{x}_i^*(t)] \quad (31)$$

and if we rewrite the collineation and the epipole as follow:

$$\mathbf{G}(t) = \begin{bmatrix} \mathbf{G}_1(t) \\ \mathbf{G}_2(t) \\ \mathbf{G}_3(t) \end{bmatrix} \quad \mathbf{e}(t) = \begin{bmatrix} \mathbf{e}_1(t) \\ \mathbf{e}_2(t) \\ \mathbf{e}_3(t) \end{bmatrix}$$

we obtain from (22), (25) and (26):

$$\begin{cases} \dot{\mathbf{G}}(t) = \dot{q}(t) [-\Phi_0 + (\mathbf{G}_0 + \Phi_0) \Gamma(t)] = \begin{bmatrix} \dot{\mathbf{G}}_1(t) \\ \dot{\mathbf{G}}_2(t) \\ \dot{\mathbf{G}}_3(t) \end{bmatrix} \\ \dot{\mathbf{e}}(t) = -\dot{q}(t) \mathbf{e}_0 = \begin{bmatrix} \dot{\mathbf{e}}_1(t) \\ \dot{\mathbf{e}}_2(t) \\ \dot{\mathbf{e}}_3(t) \end{bmatrix} \\ \alpha_i(t) = \mathbf{G}_3(t) \mathbf{x}_i^*(1) + \tau_i \mathbf{e}_3(t) \\ \dot{\alpha}_i(t) = \dot{\mathbf{G}}_3(t) \mathbf{x}_i^*(1) + \tau_i \dot{\mathbf{e}}_3(t) \end{cases}$$

The term $\frac{\partial \mathbf{s}^*}{\partial t}$ is finally obtained by introducing the previous relations in (31).

8.2 Experimental results : planar scene

The proposed method has been tested in a positioning task with respect to an unknown planar scene of a CCD camera mounted on a six degree of freedom manipulator. The extracted visual features are the image coordinates of interesting points obtained using the Harris detector [6]) and matched using the algorithm described in [16]. These points are represented by crosses in the initial and desired images (refer to Figure 8). In order to check the robustness with respect to modeling errors, we carried out two experiments. In the first one the camera parameters given by the camera manufacturer and a correct approximation of the depth d^f (i.e $d^f = 0.9$ m) have been used. In the second one an error of 30% has been added to the camera parameters, while d^f has been set to 0.5 m. As can be seen in Figures 9(a), 9(b) and 10(a), 10(b), the planned and the tracked trajectories are similar. The trajectories obtained with correct parameters are close to those obtained with bad parameters (compare Figure 9(a) with Figure 10(a) and Figure 9(b) with Figure 10(b)). That confirms the robustness of the path planning and of the control scheme with respect to calibration errors and errors on d^f . The tracking error (defined as $\frac{1}{2n} \sum_1^n (x_i(t) - x_i^*(t)) + (y_i(t) - y_i^*(t))$) and plotted in Figures 9(d) and 10(d) remains small in the two cases (less than 4 pixels). Finally, The convergence of the error on the coordinates between the current and the desired images (defined as $\frac{1}{2n} \sum_1^n (x_i(t) - x_i^*(1)) + (y_i(t) - y_i^*(1))$) and plotted in Figures 9(c) and 10(c) to 0 demonstrates the correct realization of the positioning task in the two cases.

8.3 Experimental results: non-planar scene

In order to check the validity of our approach for non-planar scene, the target is now composed by nine white marks. The extracted visual features are the image coordinates of the center of gravity of each mark. The algorithm proposed in [10] has been used to obtain the initial collineation related to the reference plane chosen as composed of the three boxed points in the Fig. 11(b). The same values of intrinsic parameters that in the case of a planar scene with bad calibration are used. The depth d^f has been set to 100cm (the real value is 50cm). In this part, we only exhibit the results since the comments in the case of an unknown planar object are also valid here. However, we note, once again, that the results is satisfactory.

9 Conclusion

In this paper, we have addressed the problem of finding the trajectories in the image space of visual features so that the camera acceleration is minimal. The obtained camera trajectory corresponds to a minimum geodesic in $SE(3)$. The method is model-free and the uncalibrated cases have been studied. By coupling the path planning step with an image-based servoing, the proposed method improves significantly the behavior of image-based servoing. We have validated our approach in a robotic platform by realizing positioning tasks with respect to an unknown scene. Future work will be devoted to introduce constraints in order to keep the target

in the camera field of view or to avoid the robot joint limits during the servoing step, whatever the displacement that has to be realized.

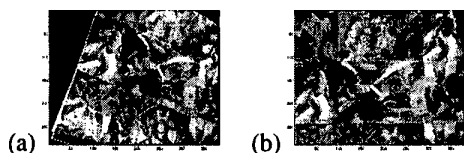


Figure 8: (a) Initial and (b) desired image

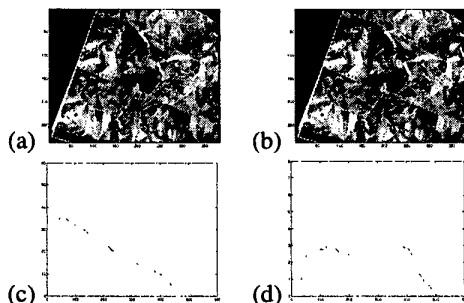


Figure 9: Correct calibration case (a) planned trajectories, (b) followed image, (c) error in image points coordinates (pixels), (d) tracking errors (pixels)

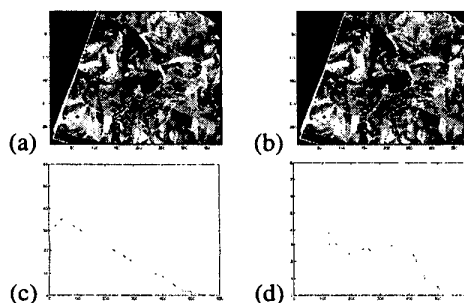


Figure 10: Bad calibration case (a) planned trajectories, (b) followed image, (c) error in image points coordinates (pixels), (d) tracking errors (pixels)

References

- [1] F. Chaumette. Potential problems of stability and convergence in image-based and position-based visual servoing. *The Confluence of Vision and Control*, D. Kriegman, G. Hager, A. Morse (eds), LNCIS Series, Springer Verlag, 237:66–78, 1998.
- [2] G. Chesi, E. Malis, and R. Cipolla. Automatic segmentation and matching of planar contours for visual servoing. In *International Conference on Robotics and Automation*, San Francisco, April 2000.
- [3] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Trans. on Robotics and Automation*, 8(3):313–326, June 1992.
- [4] O. Faugeras. *Three-dimensionnal computer vision : a geometric view point*. MIT press, Cambridge, Massachusetts, 1993.
- [5] O. Faugeras and F. Lustman. Motion and structure from motion in a piecewise planar environment. *Int. Journal of Pattern Recognition and Artificial Intelligence*, 2(3):485–508, 1988.
- [6] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Conference*, pages 189–192, 1988.
- [7] K. Hashimoto. *Visual Servoing : Real Time Control of Robot Manipulators Based on Visual Sensory Feedback*. World Scientific Series in Robotics and Automated Systems, Vol 7, World Scientific Press, Singapore, 1993.
- [8] K. Hosoda, K. Sakamoto, and M. Asada. Trajectory generation for obstacle avoidance of uncalibrated stereo visual servoing without 3d reconstruction. *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 1(3):29–34, August 1995.
- [9] S. Hutchinson, G.D. Hager, and P.I. Corke. A tutorial on visual servo control. *IEEE Trans. on Robotics and Automation*, 12(5):651–670, October 1996.
- [10] E. Malis and F. Chaumette. 2 1/2 d visual servoing with respect to unknown objects through a new estimation scheme of camera displacement. *International Journal of Computer Vision*, 37(1):79–97, June 2000.
- [11] Y. Mezouar and F. Chaumette. Design and tracking of desirable trajectories in the image space by integrating mechanical and visibility constraints. In *IEEE Int. Conf. on Robotics and Automation*, volume 1, pages 731–736, Seoul, Korea, May 2001.
- [12] F. C Park and B. Ravani. Smooth invariant interpolation of rotations. *ACM Transactions on Graphics*, 16(3):277–295, July 1997.
- [13] A. Ruf and R. Horaud. Visual trajectories from uncalibrated stereo. *IEEE Int. Conf. on Intelligent Robots and Systems*, pages 83–91, September 1997.
- [14] C. Samson, B. Espiau, and M. Le Borgne. *Robot Control : The Task Function Approach*. Oxford University Press, 1991.
- [15] R. Singh, R. M. Voyle, D. Littau, and N. P. Papanikolopoulos. Alignment of an eye-in-hand system to real objects using virtual images. *Workshop on Robust Vision for Vision-Based Control of Motion, IEEE Int. Conf. on Robotics and Automation*, May 1998.
- [16] Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence Journal*, 78:87–119, October 1995.

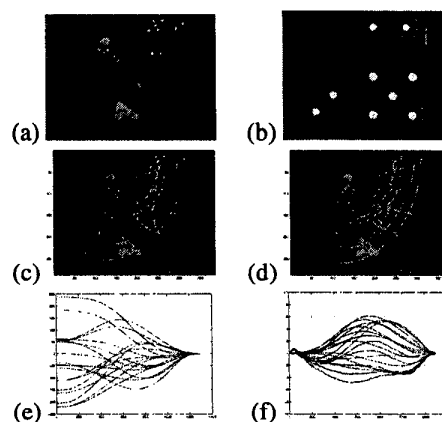


Figure 11: Experiment with a non-planar object using bad calibration: (a) initial image, (b) desired image, (c) planned trajectories, (d) followed trajectories, (e) error in image points coordinates (pixels), (f) tracking error (pixels).